

OPTIMASI HYPERPARAMETER CONVOLUTIONAL NEURAL NETWORK DENGAN ARSITEKTUR MOBILENET PADA KLASIFIKASI PENYAKIT DAUN JAGUNG

Ni Putu Dita Ariani Sukma Dewi ¹, I Gede Hendrayana ², I Wayan Agus Weda Kusuma Putra ³

^{1,2,3} Fakultas Teknologi, Program Studi Sistem dan Teknologi Informasi

Institut Teknologi dan Kesehatan Bali, Denpasar, Indonesia

ditaariani@itekes-bali.ac.id

ABSTRAK

Jagung merupakan salah satu komoditas pangan utama di Indonesia yang kaya akan karbohidrat dan Indonesia merupakan eksportir jagung terbesar di ASEAN. Produksi jagung bergantung pada kesehatan tanaman jagung. Tanaman yang terinfeksi akan menurunkan hasil produksi. Maka dari itu, penyakit pada tumbuhan menjadi tantangan besar bagi sektor pertanian, terkhususnya bagi petani jagung. Mengenali jenis penyakit pada daun jagung adalah sebuah langkah awal dalam pemberantasan dan pengobatan penyakit pada jagung sehingga dapat mengurangi kemungkinan gagal panen. Biasanya, petani jagung menggunakan cara konvensional untuk mengenali penyakit pada tanaman jagung. Namun, metode ini tidak efektif dan efisien karena memerlukan waktu yang lama dan banyak tenaga manusia. Penelitian ini akan membuat sebuah model *deep learning* untuk klasifikasi penyakit daun jagung secara otomatis dengan menggunakan salah satu arsitektur *Convolutional Neural Network* yakni *MobileNet*. Dataset yang digunakan pada penelitian ini bernama *Corn Leaf Disease* yang terdiri dari 4 kelas penyakit daun jagung yakni Bercak Daun, Hawar Daun, Karat Daun, dan Daun Sehat. Model dengan hasil terbaik menggunakan kombinasi *hyperparameter batch size* 32 dan *learning rate* 0,001. Performansi model yakni 97,83% pada *training*, 92,25% pada *validation*, dan 90,75% pada *testing*.

Keyword : Penyakit Daun Jagung; Deep Learning; Klasifikasi; Optimasi; MobileNet

1. PENDAHULUAN

Jagung merupakan tanaman yang mengandung karbohidrat tinggi yang dapat dipergunakan sebagai alternatif sumber karbohidrat yang lebih kaya serat dan protein dibandingkan nasi putih, sehingga dapat dikatakan bahwa jagung juga merupakan komoditas pangan utama selain nasi bagi masyarakat Indonesia. Sebagai komoditas pangan yang paling signifikan, jagung berada pada posisi ketiga di dunia setelah gandum dan beras, serta urutan kedua di Indonesia setelah beras. Jagung bernilai ekonomis tinggi dan manfaat yang beragam karena dapat diolah menjadi pangan, bahan baku industri, pakan ternak, dan lain sebagainya. Informasi dan data dari Badan Pangan Nasional memperlihatkan bahwa produksi jagung dalam negeri diperkirakan akan mencapai sekitar 16,84 juta ton pada tahun 2023 [1]. Persediaan awal adalah sebesar 3,29 juta ton pada tahun 2023. Permintaan jagung sepanjang tahun akan diperkirakan mencapai 16,44 juta ton [2]. Mengingat tingginya permintaan jagung tiap tahun di Indonesia, maka kualitas panen jagung juga harus lebih diperhatikan. Kualitas jagung dapat ditentukan berdasarkan kesehatan tanaman jagung. Kegagalan dalam proses pertumbuhan menyebabkan penurunan produksi jagung akibat serangan penyakit pada tanaman jagung. Badan Pusat Statistik Jawa Timur (BPS) pada tahun 2020 menyatakan, di tahun 2019 terdapat penurunan produksi jagung sebanyak 622.403 ton. Turunnya produksi tersebut diakibatkan oleh adanya penyakit yang menyerang tanaman jagung. Jika penyakit tanaman tersebut tidak segera terdeteksi dan diobati,

maka akan merusak tanaman dan menyebabkan kematian tanaman jagung [3]. Penyakit pada jagung dapat diidentifikasi dengan pengamatan yang dilakukan pada bagian tanaman yang terserang terutama daun, umumnya menunjukkan kemunculan gejala awal penyakit seperti perubahan warna dan bercak [3].

Majunya perkembangan teknologi serta ilmu pengetahuan, komputer dapat membantu identifikasi penyakit pada tanaman jagung dengan lebih akurat [4]. Pemanfaatan komputer dalam mengidentifikasi penyakit pada tanaman dapat menggunakan salah satu metode dalam kecerdasan buatan yakni *deep learning*. *Deep Learning* merupakan penggunaan arsitektur jaringan saraf tiruan yang memiliki sejumlah besar lapisan pemrosesan. Satu dari sekian metode *Deep Learning* yang umumnya dipergunakan pada klasifikasi citra adalah *Convolutional Neural Network*. *Convolutional Neural Network* adalah salah satu bagian dari *deep learning* yang mempunyai kaitan erat dengan menemukan fitur dan hubungan dalam gambar melalui urutan lapisan konvolusi, *pooling* dan *rectified linear unit* (ReLU). Lapisan-lapisan ini bekerja dengan menggabungkan berbagai fitur yang ditemukan oleh lapisan sebelumnya. CNN telah terbukti secara efektif dalam membedakan fitur terlepas dari banyaknya perubahan yang mungkin mempengaruhi gambar masukan. Dalam perancangan sistem berbasis CNN, dapat dibangun struktur jaringan dari awal atau model yang telah dilatih sebelumnya dapat digunakan kembali dan pengetahuan yang mereka pelajari dapat ditransfer

ke model lain yang dikenal sebagai *transfer learning*. *Transfer learning* mengadaptasi model yang ada dengan pelatihan ulang sebagian atau seluruhnya dengan cara yang sesuai dengan model baru. Keuntungan dari teknik *transfer learning* adalah mengenali karakteristik umum pada lapisan sebelumnya (warna, batas, dll) dan mengadaptasi lapisan berikutnya secara lebih spesifik. [5].

2. TINJAUAN PUSTAKA

Beberapa penelitian dalam deteksi dan klasifikasi mengenai penyakit jagung berbasis *deep learning* telah dilakukan. Penelitian oleh [3] mengimplementasikan CNN dengan menggunakan arsitektur *MobileNet*, *AlexNet*, dan *LeNet*. Data citra yang dipergunakan sebanyak 4000 citra yang terbagi dalam data *train* serta data *test* dengan rasio 80:20. Dari hasil eksperimen, ditemukan bahwa arsitektur *MobileNet* mengungguli *LeNet* serta *AlexNet* yang memiliki skor akurasi 83,37%, rata-rata presisi 83,37%, serta G-mean 82,98%. Nilai ini telah diverifikasi para petani jagung dan ahli pertanian yang berpengalaman di provinsi Bantul. Selanjutnya, penelitian oleh [6] mengimplementasikan CNN pada 2 kelas daun jagung yakni daun jagung sehat sejumlah 3718 citra serta daun jagung dengan penyakit karat (*common rust*) sejumlah 3814 citra. Dari hasil uji dengan rasio data 60% data *train* dan 40% sebesar 50 *epoch*, hasil akurasi yang didapatkan yakni 99,90%, nilai presisi yakni 99,81%, nilai *recall* sebesar 100%, dan nilai *F1-Score* sebesar 99,90%. Penelitian oleh [4] menerapkan model *pre-trained* CNN dalam pendeteksian jenis penyakit yang ada pada daun jagung. Deteksi tersebut dilakukan pada 5 kelas yaitu 1 daun jagung sehat serta keempat penyakit daun jagung yakni bercak daun, karat daun, bulai daun, dan karat daun. Berdasarkan hasil pengujian, terbukti penggunaan teknik *transfer learning* menimbulkan peningkatan akurasi dan mengurangi waktu komputasi. Akurasi yang didapat yakni pada data *training* sebesar 0.85 dengan *error rate* (*training loss*) sebesar 0.45 dan data validasi sebesar 0.88 dengan *error rate* (*validation loss*) sebesar 0.54. Penelitian oleh [7] menerapkan metode CNN pada klasifikasi 2 kelas penyakit tanaman jagung yakni hawar daun dan karat daun. Data citra berjumlah 2000 gambar penyakit jagung mendapatkan performansi *training* 97,5%, *validation* 100%, dan *testing* 94%.

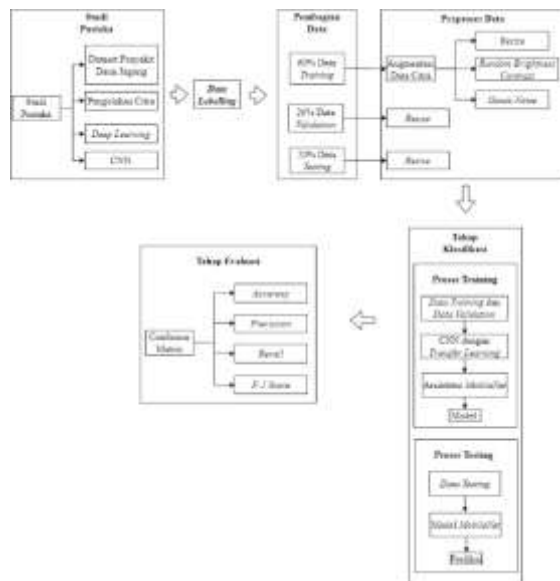
Terdapat beberapa penelitian dalam pengklasifikasian penyakit tanaman lainnya dengan menggunakan CNN, antara lain penelitian oleh [8] mengimplementasikan CNN dengan arsitektur *MobileNet* serta optimasi *hyperparameter* yang digunakan dalam proses *training* sehingga mampu menghasilkan peformansi optimal pada model CNN.

Terdiri dari 3 kelas penyakit daun padi yang diuji, diantaranya adalah *blast*, *blight* dan *tungro*. Sesuai dengan pengujian, penentuan *hyperparameter* memiliki pengaruh penting terhadap peforma model. Kombinasi *hyperparameter* yang memberikan hasil optimal adalah dengan 100 *epoch*, *batch size* 32, *learning rate* 0,001, serta *optimizer* *RMSProp* dengan nilai akurasi 97,56%, *recall* 97,57%, presisi 97,64%, dan *F1-score* 97,57%. Penelitian dari [9] menerapkan CNN dengan arsitektur VGG 16 dalam klasifikasi penyakit pada tanaman tomat yang terdiri dari 10 kelas penyakit yakni *late blight*, *septoria leaf spot*, *bacterial spot*, *early blight*, *mosaic virus*, *leaf mold*, *spider mite*, *yellow leaf virus*, dan *target spot*. Hasil yang didapatkan dengan 10519 data *training* dan 1100 data *validation* menunjukkan akurasi data *training* 98% dan 82 % untuk data *validation*. Terakhir, merupakan penelitian oleh [10] yang menunjukkan perbandingan arsitektur serta pengaruh *hyperparameter* yang digunakan. Penelitian ini menggunakan dataset TPHerbleaf dalam mengklasifikasikan 50 jenis daun tumbuhan herbal dengan perbandingan 3 arsitektur CNN yakni *Inception ResNet V2*, *MobileNet*, dan *EfficientNet B2*. *Hyperparameter* yang diuji pada penelitian ini adalah *optimizer*, *learning rate*, serta intensitas rasio penggunaan *dropout*. Performansi optimal yakni ada pada model *MobileNet* dengan *optimizer* *Adam*, *learning rate* dengan nilai 0.0001, serta *dropout* 20%. Akurasi model yang didapat adalah 100% pada *training*, 79% pada *validation*, dan 82% pada *testing*. Pada penelitian ini dibuktikan bahwa penjelasan bahwa *MobileNet* bekerja dengan keseimbangan yang menguntungkan antara kinerja dan juga ukuran model. Teknik *transfer learning* pada penelitian ini juga telah terbukti mampu mempercepat proses *training* dan menghasilkan model yang lebih baik dalam kinerjanya.

Dari beberapa studi literatur mengenai CNN tersebut, akan digunakan usulan arsitektur CNN yakni *MobileNet* dengan penyesuaian *hyperparameter* seperti *batch size* dan *learning rate*, sehingga nantinya didapatkan hasil yang optimal dalam mengklasifikasikan jenis penyakit daun jagung.

3. METODE PENELITIAN

Gambar 1 dibawah ini merupakan skema atau alur yang menjelaskan mengenai rincian tahapan penelitian awal, proses, serta hasil dari penelitian. Rincian tahapan penelitian ini mulai dari studi pustaka meliputi studi mengenai dataset dan metode, *data labelling* atau pemberian nama file di tiap kelas, pembagian data yang digunakan dalam penelitian, praproses data atau tahap persiapan data, dan tahap akhir yakni tahap klasifikasi dan evaluasi kinerja model.



Gambar 1. Alur Penelitian

3.1. Studi Pustaka

Studi pustaka pada penelitian ini terbagi menjadi 2, yakni studi pustaka mengenai sumber dataset yang digunakan dan studi pustaka literatur terdahulu yang memiliki keterkaitan dengan penelitian ini. Dataset yang dipergunakan pada penelitian ini bersumber dari Kaggle oleh [11]. Dataset tersebut merupakan dataset penyakit daun tanaman jagung yang terdiri dari 4 kelas, 3 diantaranya adalah kelas penyakit yakni bercak daun, hawar daun, karat daun, dan 1 kelas lagi adalah daun jagung yang sehat. Data yang digunakan terdiri dari data citra RGB dengan total 4000 citra, dengan masing-masing kelas berisi 1000 data citra. Data citra diambil menggunakan kamera ponsel dengan resolusi 16 MegaPixels. Gambar diambil menggunakan perlakuan khusus dengan menempatkan kertas putih di belakang daun jagung untuk memudahkan klasifikasi dan menghindari gangguan dari objek lain. Gambar diambil 5 kali untuk setiap daun jagung. Pemotretan dilakukan pada siang hari (12.00 – 14.00 WIB) untuk memastikan kondisi pencahayaan yang baik. Pelabelan data dilakukan oleh perwakilan Dinas Hortikultura dan Pertanian Perkebunan Kabupaten Sampang, dan data tersebut divalidasi dengan surat pernyataan yang ditandatangani oleh koordinator POPT (Pengamat Organisme Pengganggu Tanaman) di Kabupaten Sampang.

3.2. Pembagian Data

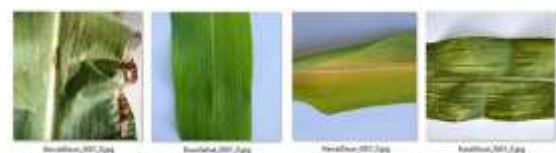
Tahapan berikutnya yakni membagi data menjadi set *training*, set *validation*, dan set *testing*. Data *training* nantinya dipergunakan untuk pelatihan model serta membantu menemukan fitur dalam data, data *validation* dipergunakan untuk penilaian model selama proses *training* untuk menghindari *overfitting*, serta data *testing* dipergunakan dalam pengujian model setelah

menyelesaikan proses *training* dan penilaian tingkat kinerja model pada data baru dalam pengujian [12]. 4000 citra daun jagung tersebut dibagi menjadi 60% untuk data *training* (2400 citra), 20% data *validation* (800 citra), dan 20% data *testing* (800 citra). Pada tiap kelas memiliki 600 citra pada data *training*, 200 citra pada data *validation*, dan 200 citra pada data *testing* dengan masing-masing 4 kelas.

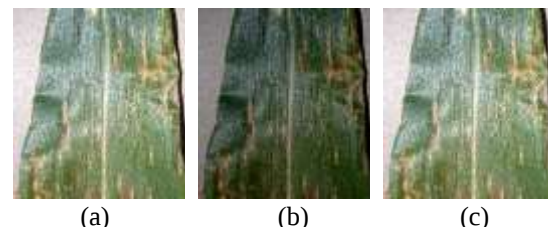
3.3. Praproses Data

Tahapan usai pembagian data yakni praproses data yang dimulai dari augmentasi data. Penerapan teknik augmentasi pada penelitian ini bertujuan guna meningkatkan kemampuan model agar mampu mempelajari objek yang sama tetapi dengan kondisi serta arah yang berbeda serta peningkatan performansi model dalam prediksi gambar baru [13]. Augmentasi pada penelitian ini menggunakan teknik *resize* untuk mengatur agar citra secara keseluruhan memiliki dimensi yang sama yakni 224x224 piksel, *random brightness contrast*, dan *gauss noise* [14].

Proses augmentasi dilakukan pada 2400 citra data *training* (600 citra pada tiap kelas), sehingga data *training* jumlahnya akan meningkat menjadi 7200 citra dengan 1800 citra pada tiap kelas, sedangkan untuk data *validation* dan *testing* hanya dilakukan teknik *resize* menjadi 224x224 piksel. Penerapan ini dilakukan agar data *validation* dan *testing* masih punya karakteristik yang sama dengan data asli, guna mampu digunakan untuk mengukur performansi model secara akurat [15]. Jumlah citra pada data *validation* dan data *testing* masing-masing masih 800 gambar (200 citra pada tiap kelas). Pada Gambar 2 berikut adalah citra sample pada masing-masing kelas penyakit daun tanaman jagung yang digunakan dalam penelitian ini, serta pada Gambar 3 adalah implementasi dari teknik augmentasi yang dilakukan.



Gambar 2. Kelas Penyakit Daun Jagung

Gambar 3. Implementasi Teknik Augmentasi *Resize* (a), *Random Brightness Contrast* (b), dan *Gauss Noise* (c)

3.4. Penerapan Model *Convolutional Neural Network*

Tahapan usai dilakukannya praproses data ialah penentuan struktur model CNN yang akan dipergunakan pada proses klasifikasi penyakit daun tanaman jagung. Pada penelitian ini akan digunakan arsitektur *MobileNet pre-trained model*, dimana *pre-trained model* mempergunakan *transfer learning* yakni model yakni telah terlatih lebih dulu dengan mempergunakan bobot dari dataset lain misalnya dengan menggunakan *ImageNet* [16]. Arsitektur *MobileNet* adalah salah satu arsitektur yang mampu mengatasi keterbatasan *computing resource*, ringan, dan memiliki latensi rendah serta dapat digunakan pada *smartphone*. *Convolution layer* pada *MobileNet* mempergunakan ketebalan kernel yang disesuaikan dengan ketebalan gambar masukan. *Depthwise convolution* dan *pointwise convolution* merupakan 2 konvolusi yang digunakan pada arsitektur *MobileNet*, dimana *depthwise separable convolution* didasarkan pada konvolusi secara mendalam yang dapat dibagi menjadi bentuk konvolusi yang melakukan penguraian konvolusi standar menjadi *depthwise convolution* dan *pointwise convolution*. *Depthwise convolution* menerapkan 1 filter pada tiap saluran masukan, lalu *pointwise convolution* mempergunakan konvolusi dengan ukuran kernel 1x1 guna menyatukan hasil *output* dari *depthwise convolution*. Pembagian lapisan inilah yang diterapkan guna dapat mengurangi komputasi dan ukuran model [17]. Berikut merupakan struktur model *MobileNet* yang digunakan dalam penelitian ini.

Model: "CornLeafDisease_MobileNet"

Layer (type)	Output Shape	Param #
input_3 (InputLayer)	[(None, 224, 224, 3)]	0
mobilenet_1.00_224 (Functional)	(None, 7, 7, 1024)	3228864
global_average_pooling2d_1 (GlobalAveragePooling2D)	(None, 1024)	0
batch_normalization_1 (Batch Normalization)	(None, 1024)	4096
dropout_1 (Dropout)	(None, 1024)	0
dense_1 (Dense)	(None, 4)	4100

 Total params: 3237060 (12.35 MB)
 Trainable params: 6148 (24.02 KB)
 Non-trainable params: 3230912 (12.32 MB)

Gambar 4. Struktur Model *MobileNet*

Berdasarkan struktur model *MobileNet* usulan pada Gambar 4, usulan arsitektur *MobileNet* dalam penelitian ini tersusun atas beberapa bagian. Langkah awal adalah citra masukan dengan dimensi 224x224 piksel dan 3 *channel* RGB. Lalu *MobileNet* (*MobileNet 1.00 224*) yakni layer *pre-trained model MobileNet* yang bertugas mengambil fitur gambar sebagai *feature extractor* dan *output* yang dihasilkan berupa fitur berukuran 7x7 dengan 1024 *channel*. Selanjutnya adalah *Global Average Pooling* untuk perubahan *layer* sebelumnya ke *output* akhir menjadi fitur vektor dengan ukuran 1024, dengan menggunakan rata-rata fitur keseluruhan unit spasial dan mengubah *output layer* sebelumnya menjadi fitur vektor dengan ukuran 1024. Selanjutnya yakni *Batch Normalization*, yang berfungsi dalam percepatan proses *training* dan peningkatan stabilitas dengan ukuran vektor fitur tetap 1024 [18]. Setelahnya, penambahan *layer Dropout*, dipergunakan guna mencegah *overfitting* dengan menonaktifkan beberapa neuron secara acak selama proses *training* dan *outputnya* berukuran 1024. *Dense (Fully Connected Layer)* dengan 4 neuron. *Layer* ini berperan menghasilkan vektor dengan ukuran 4 sebagai prediksi akhir, sesuai banyaknya kelas yang nantinya dilakukan prediksi. Adanya penetapan nilai *hyperparameter* sangat diperlukan yakni seperti *early stopping*, *loss function*, *epoch*, dan *optimization function*, serta metrik-metrik pengujian yang dipergunakan sebelum dimulainya proses *training*.

Saat proses pelatihan dilakukan, model akan mempelajari keseluruhan data melalui jumlah *epoch* yang ditentukan, yakni 100 *epoch*. Penerapan penggunaan *early stopping* pada model ini dimaksudkan untuk memastikan bahwa model tidak harus menempuh *epoch* secara keseluruhan untuk mencapai konvergensi. Maka, jika nilai *validation loss* tidak berkurang maka proses pelatihan akan langsung terhenti. Pada hal ini, *patience* bernilai 10 dipergunakan dalam penentuan batas *epoch* berikutnya jika nilai *validation loss* tidak mengalami penurunan sama sekali (stagnan). *Early stopping* umum dipergunakan dalam penghematan sumber daya. Selain itu, terdapat *hyperparameter* lain yang harus ditentukan terlebih dahulu seperti *batch size*, *learning rate*, dan *optimizer*. Detail dari nilai *hyperparameter* serta skenario pengujian yang dilakukan pada penelitian ini adalah sebagai berikut.

Tabel 1. Detail Skenario Pengujian dan Nilai *Hyperparameter*

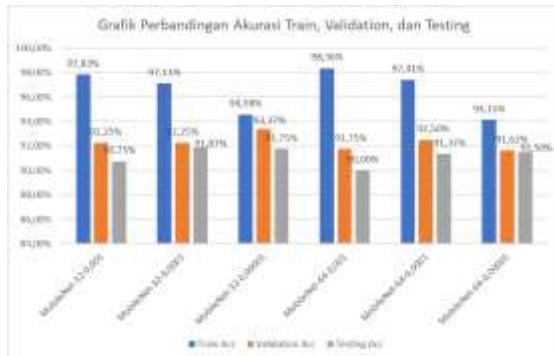
Skenario	Arsitektur	Batch Size	Optimizer	Dropout	Learning Rate	Max Epoch/patience
1	Mobile Net	32	Adam	20%	0,001	100/10
2	Mobile Net	32	Adam	20%	0,0001	100/10
3	Mobile Net	32	Adam	20%	0,00001	100/10
4	Mobile Net	64	Adam	20%	0,001	100/10
5	Mobile Net	64	Adam	20%	0,0001	100/10
6	Mobile Net	64	Adam	20%	0,00001	100/10

4. HASIL DAN PEMBAHASAN

Berikut merupakan hasil dan pembahasan dari penelitian yang dilakukan.

4.1. Pengujian Model

Uji coba model dilakukan pada *Google Colaboratory* dengan menggunakan *Keras (Python Deep Learning Library)*, serta menggunakan layanan GPU. Ilustrasi grafik dari hasil akurasi *training*, *validation*, dan *testing* untuk ke-6 skenario uji coba model MobileNet dapat ditemukan dalam Gambar 5 sebagai berikut.



Gambar 5. Grafik Perbandingan Akurasi 6 Skenario Model MobileNet

Detail nilai serta penentuan *ranking* dari model terbaik ditunjukkan pada Tabel 2, penentuan model terbaik didasarkan pada *Cost Benefit Analysis*. *Cost Benefit Analysis* adalah suatu pendekatan yang

digunakan dalam penentuan dan perhitungan nilai dari tiap-tiap komponen teknologi informasi yang mempunyai pengaruh terhadap biaya (*cost*) yang dikeluarkan serta keuntungan yang didapat (*benefit*). Dalam penentuan model terbaik, *test accuracy* adalah parameter utama pada *benefit*, sedangkan parameter utama dari *cost* adalah *training time* [19]. Nilai pada penentuan peringkat menggunakan teknik normalisasi yakni pada Persamaan (1).

$$r_{ij} = \begin{cases} \frac{x_{ij}}{\max_i(x_{ij})} & \text{Jika } X_{ij} \text{ merupakan parameter cost} \\ \frac{\min_i(x_{ij})}{x_{ij}} & \text{Jika } X_{ij} \text{ merupakan parameter benefit} \end{cases} \quad (1)$$

Nilai x_{ij} merupakan parameter *cost* atau *benefit*, r_{ij} adalah parameter yang dilakukan normalisasi, i merupakan alternatif, dan j merupakan parameter. Lalu, proses selanjutnya adalah penjumlahan (*summation*) menggunakan Persamaan (2) sebagai berikut.

$$V_i = \sum_{j=1}^n W_j r_{ij} \quad (2)$$

Nilai V_i merupakan nilai penjumlahan bobot untuk tiap alternatif, W_j merupakan nilai bobot pada tiap parameter, dan bobot yang dipergunakan pada perhitungan ini adalah 1 untuk semua kriteria [20].

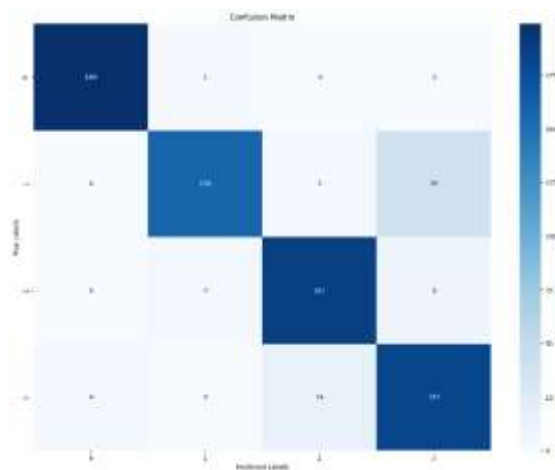
Tabel 2. Detail Hasil Pengujian Model dan *Cost Benefit Analysis*

Skenario	Test Acc	Training Time (s)/Epoch	Benefit Norm (r_{i1})	Cost Norm (r_{i2})	Sum (V_i)	R
1	90,75%	2770 / 13	0,988	1,000	1,988	1
2	91,87%	4869 / 29	1,000	0,569	1,569	2
3	91,75%	5505 / 85	0,999	0,503	1,502	3
4	90,00%	5858 / 17	0,980	0,473	1,453	5
5	91,37%	5559 / 48	0,995	0,498	1,493	4
6	91,50%	6113 / 100	0,996	0,453	1,449	6

Berdasarkan nilai ke-6 pengujian model diatas, ditemukan hasil terbaik yakni model arsitektur MobileNet (Skenario 1) dengan *batch size* 32 dan *learning rate* 1×10^{-3} atau 0,001 dimana model tersebut memiliki nilai akurasi sebesar 97,83% untuk *training* dan 92,25% untuk *validation*, serta menurut analisis *cost and benefit* model yang telah dilakukan, ditemukan bahwa skenario 1 unggul dalam akurasi *testing* yakni 90,75% dan waktu *training* tercepat yang menandakan bahwa model pada skenario 1 adalah model yang paling optimal dari segi efisiensi model. Setelah melalui proses pencarian yang cermat, model terbaik untuk klasifikasi penyakit daun jagung telah ditemukan. Langkah selanjutnya adalah menyelami performa model ini dengan lebih detail. *Confusion Matrix* menjadi alat bantu utama dalam memberikan gambaran menyeluruh tentang keberhasilan

klasifikasi pada dataset yang digunakan. Sebuah matriks hasil dari prediksi nantinya dilakukan perbandingan dengan kelas sebenarnya atau bisa dikatakan bahwa matriks ini memuat informasi nilai sebenarnya (*actual value*) dan nilai prediksi (*prediction value*) pada klasifikasi. *Confusion Matrix* menjadi landasan untuk menghitung metrik-metrik kunci yang mencerminkan performa model secara komprehensif seperti akurasi, presisi, *recall*, dan *f1-Score*. Akurasi menunjukkan proporsi prediksi yang benar secara keseluruhan. Nilai akurasi yang tinggi menunjukkan bahwa model mampu mengklasifikasikan data dengan baik secara umum. Presisi mengukur proporsi prediksi positif yang benar-benar positif. Nilai presisi yang tinggi menunjukkan bahwa model dapat meminimalisir kesalahan positif (positif palsu). *Recall* menunjukkan proporsi data aktual yang positif yang

berhasil diidentifikasi oleh model sebagai positif. Nilai *recall* yang tinggi menunjukkan kemampuan model dalam menangkap semua kasus positif yang ada. Terakhir, *F1-score* merupakan gabungan presisi dan *recall*, memberikan gambaran seimbang tentang kemampuan model dalam mengidentifikasi data positif secara akurat [10]. Berikut dibawah ini merupakan hasil analisis dan *confusion matrix* dari model terbaik yang telah dilatih dan telah dilakukan proses pengujian (*testing*). Label 0 mewakili kelas BercakDaun, 1 mewakili kelas DaunSehat, 2 mewakili kelas HawarDaun, dan 3 mewakili kelas KaratDaun.



Gambar 6. Confusion Matrix Skenario 1 MobileNet (Testing)

Berdasarkan *confusion matriks* pada Gambar 6, terlebih dahulu menentukan nilai *total true positive* (TTP), *total false positive* (TFP), serta *total false negative* (TFN) sebelum menghitung nilai metrik akurasi, *recall*, presisi, dan *f-1 score*. Berikut adalah perhitungan dari keempat metrik yang telah ditentukan.

TTP : 726

TFP : 74

TFN : 74

Total Akurasi

$$\begin{aligned}
 &= \frac{TTP_{all}}{\text{Jumlah Data Testing}} \times 100\% \\
 &= \frac{726}{800} \times 100\% \\
 &= 90,75\%
 \end{aligned}$$

$$\begin{aligned}
 \text{Rerata Presisi} &= \frac{\sum \text{Presisi}}{\text{Jumlah Kelas}} \times 100\% \\
 &= \frac{3,66}{4} \times 100\% \\
 &= 91,5\%
 \end{aligned}$$

$$\begin{aligned}
 \text{Rerata Recall} &= \frac{\sum \text{Recall}}{\text{Jumlah Kelas}} \times 100\% \\
 &= \frac{3,63}{4} \times 100\% \\
 &= 90,75\%
 \end{aligned}$$

$$\begin{aligned}
 F-1 \text{ Score} &= \frac{2 (\text{Presisi} \times \text{Recall})}{\text{Presisi} + \text{Recall}} \% \\
 &= \frac{2 \times (91,5 \times 90,75)}{(91,5 + 90,75)} \% \\
 &= 91,12\%
 \end{aligned}$$

Berdasarkan perhitungan *confusion matrix*, diketahui bahwa model ini memiliki kinerja yang baik. Secara spesifik, model tersebut mampu mengenali objek dengan benar sebesar 90,75% atau 726 citra dari total 800 citra, objek yang salah terdeteksi sebesar 74 citra, dan gagal mendeteksi objek sebanyak 74 citra. Dari 74 citra yang salah dideteksi tersebut merupakan objek yang seharusnya terdeteksi sebagai objek lain (*false positive*). Sedangkan dari 74 citra yang gagal terdeteksi tersebut adalah objek yang seharusnya terdeteksi (*false negative*). Nilai presisi model ini sebesar 91,5%, yang artinya model mampu memprediksi objek sebagai objek yang benar sebanyak 91,5% dari total objek yang diprediksi sebagai objek yang benar. Nilai *recall* model ini sebesar 90,75%, yang artinya model mampu mendeteksi sebanyak 90,75% dari total objek yang benar. Nilai *F1-score* model ini sebesar 91,12%, yang merupakan rata-rata nilai dari presisi dan *recall* dan memperlihatkan jika model memiliki performa yang baik dalam hal presisi dan *recall*.

4.2. Pengujian Model

Setelah mendapatkan model dengan akurasi terbaik, langkah selanjutnya adalah menyimpan model tersebut dalam format file .h5. File ini kemudian digunakan untuk melakukan klasifikasi pada data baru melalui laman *website* yang telah dirancang. Pengguna dapat mengunggah data baru ke *website*, dan data tersebut akan diproses oleh model yang telah dimuat. Model akan memproses data baru dan menghasilkan prediksi mengenai klasifikasi penyakit daun jagung. Hasil prediksi ini kemudian ditampilkan kepada pengguna dalam bentuk visual yang mudah dipahami, yakni grafik persentase. Grafik ini menunjukkan persentase kemungkinan data baru termasuk dalam kelas tertentu. Penyajian hasil prediksi dalam bentuk grafik persentase ini memudahkan pengguna untuk menginterpretasikan hasil dengan lebih baik. Pengguna dapat dengan cepat melihat kelas mana yang memiliki kemungkinan paling tinggi dengan data baru yang telah di unggah. Contoh dari grafik persentase hasil prediksi ditunjukkan pada Gambar 7.



Gambar 7. Website Klasifikasi Penyakit Daun Jagung

Berdasarkan ilustrasi pada Gambar 7, pengguna atau *user* akan mengunggah data uji baru ke laman *website* dan hasil dari prediksi model akan ditampilkan pada laman *website* dalam bentuk persentase kelas yang sesuai.

5. KESIMPULAN DAN SARAN

Penelitian ini melakukan optimasi *hyperparameter Convolutional Neural Network* menggunakan arsitektur *MobileNet* dalam mengklasifikasikan penyakit daun jagung. Optimasi *hyperparameter* yang dilakukan diantaranya ialah pada *batch size*, dan *learning rate*. Model ini dilatih dengan dataset *Corn Leaf Disease* dan menghasilkan akurasi terbaik dengan kombinasi *batch size* 32 dan *learning rate* 0,001. Hasilnya, model dapat mencapai performansi yang optimal yakni dengan akurasi 97,83% pada *training*, 92,25% pada *validation*, dan 90,75% pada *testing*. Berdasarkan *Cost Benefit Analysis*, kelebihan yang ditemukan pada model ini yakni memiliki akurasi tinggi serta ukuran model yang relatif kecil sehingga dapat menghemat sumber daya serta biaya komputasi. Pengembangan penelitian selanjutnya yang dapat dilakukan yakni dengan penyesuaian parameter seperti mencari kombinasi *epoch*, *learning rate*, dan *optimizer* yang optimal untuk meningkatkan performa model, penyesuaian arsitektur model seperti menambahkan atau mengatur ulang *layer*, seperti *dropout* dan *batch normalization* untuk meningkatkan kemampuan model dalam menggeneralisasi data baru. Selain itu, adapun pengembangan aplikasi model dapat dilakukan pada *smartphone* dalam bentuk aplikasi *mobile* agar proses uji coba dapat dilakukan secara langsung (*realtime*). Dengan pengembangan yang berkelanjutan, model ini diharapkan dapat menjadi alat yang efektif dan mudah diakses bagi petani dalam mendeteksi penyakit daun jagung secara dini dan akurat. Hal ini dapat membantu meningkatkan hasil panen dan mengurangi kerugian ekonomi akibat penyakit tanaman.

DAFTAR PUSTAKA

- [1] N. Putra, "Analysis of the Effect of Harvest Area, Agricultural Credit, Labor, on Food Production (Rice and Corn) in North Sumatra," *Inspirat*, vol. 12, no. 1, pp. 12–22, 2022.
- [2] F. Rozi *et al.*, "Indonesian market demand

patterns for food commodity sources of carbohydrates in facing the global food crisis," *Heliyon*, vol. 9, no. 6, p. www.sciencedirect.com/science/article/pii/S2405844, 2023.

- [3] A. Y. Pratama and Y. Pristyanto, "CLASSIFICATION OF CORN PLANT DISEASES USING VARIOUS CONVOLUTIONAL NEURAL NETWORK," *JITK J. Ilmu Pengetah. dan Teknol. Komput.*, vol. 9, no. 1, 2023, [Online]. Available: <https://ejournal.nusamandiri.ac.id/index.php/jitk/article/view/4258/1067>
- [4] M. I. Rosadi and M. Lutfi, "Identifikasi Jenis Penyakit Daun Jagung Menggunakan Deep Learning Pre-Trained Model," *J. Explor. IT!*, vol. 13, no. 2, pp. 35–42, 2021, [Online]. Available: <https://jurnal.yudharta.ac.id/v2/index.php/EXPLORE-IT/article/view/2690/1963>
- [5] M. Fraiwan, E. Faouri, and N. Khasawneh, "Classification of Corn Diseases from Leaf Images Using Deep Transfer Learning," *Plants (Basel)*, vol. 11, no. 20, 2022, doi: 10.3390/plants11202668.
- [6] A. B. Prakosa, Hendry, and R. Tanone, "IMPLEMENTASI MODEL DEEP LEARNING CONVOLUTIONAL NEURAL NETWORK (CNN) PADA CITRA PENYAKIT DAUN JAGUNG UNTUK KLASIFIKASI PENYAKIT TANAMAN," *J. Pendidik. Teknol. Inf.*, vol. 6, no. 1, pp. 107–116, 2023, [Online]. Available: <https://ojs.cbn.ac.id/index.php/jukanti/article/view/919/340>
- [7] D. Iswanto and D. H. UN, "Klasifikasi Penyakit Tanaman Jagung Menggunakan Metode Convolutional Neural Network (CNN)," *J. Ilm. Univ. Batanghari Jambi*, vol. 22, no. 2, pp. 900–905, 2022, doi: 10.33087/jiubj.v22i2.2065.
- [8] A. Julianto, A. Sunyoto, and F. W. Wibowo, "OPTIMASI HYPERPARAMETER CONVOLUTIONAL NEURAL NETWORK UNTUK KLASIFIKASI PENYAKIT TANAMAN PADI," *TEKNIMEDIA*, vol. 3, no. 2, 2022, [Online]. Available: <https://jurnal.stmiksznw.ac.id/index.php/teknimedia/article/view/77>
- [9] R. Soekarta, N. Nurdjan, and A. Syah, "Klasifikasi Penyakit Tanaman Tomat Menggunakan Metode Convolutional Neural Network (CNN)," *J. INSECT*, vol. 8, no. 2, pp. 143–151, 2023, [Online]. Available: <https://ejournal.um-sorong.ac.id/index.php/insect/article/view/2356/1331>
- [10] N. P. D. A. S. Dewi, M. W. A. Kesiman, I. M. G. Sunarya, I. G. A. A. D. Indradewi, and I. G.

- Andika, "Klasifikasi Jenis Daun Tumbuhan Herbal Berdasarkan Lontar Usada Taru Pramana Menggunakan CNN," *J. Techno.com*, vol. 23, no. 1, pp. 271–283, 2024, [Online]. Available: <https://publikasi.dinus.ac.id/index.php/technoc/article/view/9510>
- [11] S. I. Saputra, "Corn Leaf Disease," Kaggle. Accessed: Apr. 10, 2024. [Online]. Available: <https://www.kaggle.com/datasets/ndisan/corn-leaf-disease>
- [12] D. E. Birba, "A Comparative study of data splitting algorithms for machine learning model selection," KTH Royal Institute Of Technology, 2020. [Online]. Available: <https://www.diva-portal.org/smash/get/diva2:1506870/FULLTEXT01.pdf>
- [13] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *J. Big Data*, vol. 6, no. 60, pp. 1–48, 2019, doi: 10.1186/s40537-019-0197-0.
- [14] K. Alomar, H. I. Aysel, and X. Cai, "Data Augmentation in Classification and Segmentation: A Survey and New Strategies," *J. Imaging*, vol. 9, no. 2, p. 46, 2023, doi: 10.3390/jimaging9020046.
- [15] F. Chollet, *Deep Learning with Python*. New York: Manning Publications Co., 2018.
- [16] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," *arXiv:1905.11946*, 2019, doi: 10.48550/arXiv.1905.11946.
- [17] A. F. Jauhari, "KLASIFIKASI JENIS BERAS MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK PADA ARSITEKTUR MOBILENET," UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM, 2022. [Online]. Available: <http://etheses.uin-malang.ac.id/37318/7/15650067.pdf>
- [18] J. Bjorck, C. Gomes, B. Selman, and K. Q. Weinberger, "Understanding Batch Normalization," in *32nd Conference on Neural Information Processing Systems (NeurIPS 2018)*, Montréal, Canada, 2018. doi: 10.48550/arXiv.1806.02375.
- [19] G. I. R. Martha, "INVESTMENT ANALYSIS: PENINGKATAN PERFORMA DAN EFFISIENSI DARI ADOPTSI CLOUD ENGINE PADA INDUSTRI KREATIF BERBASIS DIGITAL VISUAL DESAIN YANG MEMILIKI KEBUTUHAN KOMPUTASI & PROSESING TINGGI," INSTITUT TEKNOLOGI SEPULUH NOPEMBER, 2019.
- [20] I. M. A. Wirawan, R. Wardoyo, D. Lelono, and S. Kusrohmaniah, "Continuous Capsule Network Method for Improving Electroencephalogram-Based Emotion Recognition," *Emerg. Sci. J.*, vol. 7, no. 1, p. 125, 2023, [Online]. Available: <https://ijournalse.org/index.php/ESJ/article/view/1269>